REMARKS

Reconsideration and allowance are respectfully requested. Claims 1-13 are currently pending. Claims 1-13 were rejected. Claim 13 has been amended to address a §112 rejection raised in the last Office Action. No new matter has been entered. Also, Claims 16 and 17 are hereby added and consideration thereof is requested. Based on the following remarks, it is believed that all of pending Claims 1-13, 16 and 17 are in condition for allowance and a notice to that affect is respectfully requested.

I. Revisions to the Detailed Description

The detailed description of the invention has been amended to include section headings. No new matter has been entered.

II. §112 Rejection of Claim 1

Claim 1 was rejected under 35 U.S.C. §112 as being indefinite. Specifically, the Office Action inquired as to where the storage is in step "d". The Office Action also noted that it is not clear as to what happens to the image data in step "i". Amended Claim 1 now addresses these rejections, specifying in step "d" that the image data and shading data is stored in a local memory, and specifying in step "i" that the shading data is provided to a frame buffer memory.

III. §103(a) Rejection of Claims 1-13

Claims 1-13 were rejected under 35 U.S.C. §103(a) as being unpatentable over U.S. Patent No. 5,852,443 to Kenworthy. Based on the following remarks, Applicant believes that this rejection is overcome.

A. Independent Claims 1 and 4

The present invention concerns the generating of 3-D computer graphics, wherein an image plane is subdivided into sub-regions or tiles which can be processed in turn. In general, this rendering of an image involves identifying the surfaces that define the various objects that are to be displayed in the image. Data defining these surfaces are stored in a type of memory known as a display list, which avoids the need to store identical surfaces for each tile, since one object made of many surfaces could appear in a number of tiles. In order to display an image, each tile is processed in sequence by passing the surface-defining data contained in the display list onto an image synthesis processor (ISP) and a texture and shading processor (TSP). The ISP and TSP subsequently process or render this data and passes it on to a frame buffer memory, where it can then be used to generate an image on a display.

The complexity of scenes or images that are expected to be rendered by 3-D graphic systems is increasing as computer hardware becomes more powerful. However, this presents a problem for display list-based graphic systems such as that generally described above because the display list storage requirements increases significantly. This typically necessitates an increase in the amount of memory in the system, which unfortunately is often cost prohibited.

To address the above problem, Applicant developed a method and system that makes optimum use of memory while rendering complex images. As called for in amended Claim 1, this method calls for a sequence of steps, including:

b) loading object data for each rectangular area into a display list memory until that memory is substantially full;

c) deriving image data and shading data for each picture element of each rectangular area from the object data;

d) <u>storing the image data and the shading data in a local memory;</u>

e) <u>loading further object data into the display list memory</u> to replace the existing contents;

f) <u>retrieving the stored image data and shading data;</u>

g) <u>deriving additional image data and shading data</u> for each picture element of each rectangular area <u>from the new object data and the previously derived image data and shading data</u>

(emphasis added).

Unlike the claimed invention, Kenworthy does not disclose a method for optimizing a display list memory whereby object data stored in a display list is partially rendered and then stored in a local memory so as to free up space in the display list memory, thereby allowing additional object data to be added to the display list memory and rendered in conjunction with the previously partially rendered data. Instead, Kenworthy discloses a significantly different method whereby the processing or rendering of object data for a tile is repeatedly abandoned and restarted.

Specifically, Kenworthy divides an image or scene into a plurality of tiles (referred to as "chunks" in Kenworthy). Each tile is sequentially processed, whereby data from a tile is sent to a rasteriser (similar to the TSP in the present invention) and then stored in pixel and fragment buffers. If the capacities of the pixel and fragment buffers are exceeded, such that these memories overflow with data, Kenworthy halts the processing of the data and goes back and divides the previously defined tile into four smaller tiles. Kenworthy then restarts the entire rendering process by selecting one of the newly defined smaller tiles and processing the data for

that tile. If the pixel and fragment buffers fill-up again for any of the newly defined smaller tiles, rendering is halted again and the smaller tile is split into four even smaller tiles. Rendering is then once again restarted from the beginning, with each of the most recently defined "sub-sub" tiles processed in sequence. (See Kenworthy, 4:41-56)

Accordingly, Kenworthy discloses a method wherein when a tile of an image to be rendered is too large or complex, such that the capacity of the memory is exceeded, the system recursively subdivides the tile, and then repeats the rendering process from the beginning for one of the subdivided tiles in hopes of avoiding another data overflow of the memory areas. Consequently, Kenworthy is a far less efficient system and method as it requires the abandonment and restarting of the processing of a tile, leading to the repeated processing of tile data.

The Applicant notes that the Office Action attempts to isolate and correlate each step of Claim 1 to an individual operation in Kenworthy. However, reference to these various individual operations in Kenworthy do not form any type of useful sequence of events similar to that called for in amended Claim 1. When considered as a whole, Kenworthy is seen to disclose a drastically different method of rendering a 3-D image whereby processing can be repeatedly halted and restarted from the beginning, resulting in a far less efficient method.

In contrast, the method called for in Claim 1 is not concerned with an internal memory whose capacity may or may not be exceeded. Instead, it addresses the situation where it is known in advance that there may not be sufficient display list memory to store the entire object data. To this end, the claimed method loads object data into the display list memory until it is substantially full, and then based on that object data, derives and stores corresponding image and shading data

into a local memory, thereby freeing up the display list memory and allowing it to be loaded with additional object data. The previously stored image and shading data is subsequently retrieved from the local memory, and then additional image and shading data is derived from the new object data and the previously derived image and shading data.

For the reasons cited above, Applicant believes that independent Claim 1, along with its dependent claims, is allowable over the cited prior art. For similar reasons, it is believed that corresponding method Claim 4, along with dependent Claims 5-6, are also allowable over the cited prior art.

B. Independent Claims 7 and 10

Independent Claim 7 calls for a memory management system for generating 3-D computer images. As called for in the claim, each of the substantially rectangular areas into which the image is divided further comprises a plurality of smaller rectangular areas. In this instance, each of the larger rectangular areas corresponds to what is referred to in the detailed disclosure as macro-tiles. Each macro-tile comprises a group of smaller rectangular areas, with each of these smaller rectangular areas representing the basic unit area of the scene or image that is processed by the present invention.

By dividing the image into macro-tiles and tiles, or larger and smaller rectangular areas, the claimed invention is able to balance the conflicting requirements of memory wastage, caused by parameter duplication when objects fall within more than one tile, with the overheads required in maintaining a large number of memory blocks in the display list memory.

In contrast, Kenworthy discloses a system that goes back and recursively divides the scene or image area into ever-decreasing smaller areas and subsequently restarts rendering

from the beginning whenever a memory overflow situation occurs. The Office Action notes that Kenworthy suggests that its image subdivision technique may be applicable to a full frame buffer, which may be decomposed into smaller regions to reduce fragment memory requirements. However, such a feature of Kenworthy neither address nor resolves issues concerning memory wastage due to parameter duplication. The Kenworthy method of recursively dividing an area into smaller areas and analyzing those smaller areas one at a time has no impact on typical situations where there is no control over the order that object data is presented to the system. Each piece of object data may be projected onto any area of the screen, and thus the system of the present invention must be able to handle a full-sized image at the frame buffer.

For the reasons cited above, Applicant believes that independent Claim 7, along with its dependent claims, is allowable over the cited prior art. For similar reasons, it is believed that corresponding method Claims 10-11 are also allowable over the cited prior art.

C. Independent Claims 12 and 13

Independent Claim 12 calls for a memory management system that comprises:

> means for subdividing the image data into a plurality of substantially rectangular areas,
>
> means for storing data pertaining to surfaces making up the image in a display list memory,
>
> means for allocating at least one block of storage from the display list memory to each rectangular area,
>
> means for storing in that block of memory data pertaining to surfaces which intersect that rectangular area,

> means for supplying data for each rectangular area from the display list to a means for deriving shading data for each picture element of the rectangular area, and
>
> frame store means for storing the shading data for display,
>
> characterised in that the means for allocating blocks of storage from the display list memory determines when a predetermined number of blocks have been used and, in dependence on the determination, causes the means for supplying data to the means for deriving shading data to commence operation, thereby releasing blocks of storage for further rectangular areas

(emphasis added).

In contrast to Claim 12, Kenworthy neither discloses nor suggests a system that divides a display list memory into blocks of storage, with at least one block of storage being allocated to each rectangular area of image data. Furthermore, Kenworthy never discloses nor suggests a system that determines when a predetermined number of storage blocks have been used, and upon such a determination, causes the deriving of shading data to commence and thus release previously held blocks of storage, which can now be used to store additional image data.

Instead, Kenworthy discloses the use of two separate pixel buffers that operate in a manner commonly known as double buffering, where pixel data output from a second chunk is written to a second buffer while some other process consumes pixel data from a first chunk by reading it from the first buffer. This consuming process occurring for the first chunk of data is completed before the second chunk of data is finished. The first buffer can then be completely cleared and reloaded with a new, third chunk of data. (See Kenworthy, 33:50-65)

Accordingly, Kenworthy does not determine when a predetermined number of storage blocks within a display list memory have been used and subsequently initiates shading so as to release previously held blocks of storage within the same display list memory. Instead, Kenworthy relies on two separate buffer memories, filling one up with a first set of data for processing, and then filling up a separate, second buffer with a second set of data for processing.

For the reasons cited above, Applicant believes that independent apparatus Claim 12, along with corresponding method Claim 13, are allowable over the cited prior art.

D. New Independent Claims 16 and 17

New independent apparatus Claim 16 and corresponding method Claim 17 call for subdividing image data into a plurality of substantially rectangular areas, with a "predefined number of neighboring rectangular areas defining a macro-area". Further called for is a display list memory that is divided into "portions corresponding to each of the defined macro-areas".

In contrast, Kenworthy neither discloses nor suggests dividing image data into a plurality of rectangular areas, where a predefined number of such neighboring areas make up a macro-area that corresponds to a select portion within the display list memory. Instead, as previously discussed, Kenworthy simply discloses the recursive division of screen areas into ever-decreasing smaller areas, and the subsequent restarting of the rendering process, when ever the capacity of a buffer memory is exceeded. For these reasons, Applicant believes that new Claims 16 and 17 are allowable over the cited prior art.

All objections and rejections having been addressed. It is respectfully submitted that the present application is in

condition for allowance, and a Notice to that effect is earnestly solicited.

Respectfully submitted,

Steven Thiel

For Mark L. Maki

MLM/SRT/cc/jas

FLYNN, THIEL, BOUTELL
& TANIS, P.C.
2026 Rambling Road
Kalamazoo, MI 49008-1631
Phone: (269) 381-1156
Fax: (269) 381-5465

Dale H. Thiel	Reg. No. 24 323
David G. Boutell	Reg. No. 25 072
Ronald J. Tanis	Reg. No. 22 724
Terryence F. Chapman	Reg. No. 32 549
Mark L. Maki	Reg. No. 36 589
Liane L. Churney	Reg. No. 40 694
Brian R. Tumm	Reg. No. 36 328
Steven R. Thiel	Reg. No. 53 685
Donald J. Wallace	Reg. No. 43 977
Kevin L. Pontius	Reg. No. 37 512
Sidney B. Williams, Jr.	Reg. No. 24 949

Encl: Post Card

136.07/05